

REMARKS

Claims 1-26 are pending in this application. The claims have not been amended further. In response to the final office action, the applicant will endeavor to describe in further details the differences between the cited references and the pending claims. In short, *Kush* describes a method of adjusting thread priorities but it does not provide reasons why the priority of the threads should be adjusted. The pending claims are directed to such reasons.

The applicant requests that the Examiner review the below statements.

Claim Rejections Under 35 USC §102

Claims 1-6, 9-15, 18-23 and 26 are rejected under 35 U.S.C. 102(b) as being anticipated by *Kush* US Patent 6,874,144 (“Kush”).

Claims 1, 10 and 19

Each of the pending independent claims call for determining whether to boost a priority of the application thread according to criteria **based on a status of I/O operations performed for the application thread** (emphasis added). *Kush* is concerned with situations where threads with a lower priority block threads with a higher priority. *Kush* does not disclose or describe a system where the decision on whether to boost a priority of the application thread is made based on the status of I/O operations performed for the application thread. *Kush* describes a mechanism for keeping threads moving through the processor and avoiding blocks. The pending application takes the methodology of *Kush* and implements it in a way that results in an improvement in usability.

Kush appears to be prone to the same problem that the pending claims are attempting to address, specifically, repeated switching between I/O threads and application threads. In *Kush*, whether a thread is an I/O thread or any other type of thread is immaterial. All that matters is the thread priorities. As this claimed element of looking to the status of I/O operations to decide whether to boost priority of the application thread is in all the independent claims and is not present in the prior art, a prima facie case of anticipation has not been made.

The Office action points to several places in *Kush* for disclosing that I/O operations are used to decide whether to boost application priorities. In the rejection, the Office action points to the Abstract; Fig. 1; Fig. 3a (mutex ID field); col. 1, lines 55-65; col. 4, lines 63-67; col. 5 lines 1-10; and col. 6, lines 10-30.

The Abstract discloses a first list for boosting a thread to an indicated priority and a second list for storing the processed boost request. It does not disclose using “determining whether to boost a priority of the application thread according to criteria based on a status of I/O operations performed for the application thread” as claimed.

Fig. 1 illustrates a network of clients, a network of printers and a print manager between the networks. While printers are a form of I/O devices, Fig. 1 does not disclose “determining whether to boost a priority of the application thread according to criteria based on a status of I/O operations performed for the application thread” as claimed.

Fig. 3a illustrates a boost request data structure, including a Mutex ID field. A mutex is a block but Fig. 3a does not illustrate “determining whether to boost a priority of the application thread according to criteria based on a status of I/O operations performed for the application thread” as claimed.

Col. 1, lines 55-65 describes how a mutex operates and how priority works with respect to threads. It does not describe “determining whether to boost a priority of the application thread according to criteria based on a status of I/O operations performed for the application thread” as claimed.

Col. 4, lines 63-67 and Col. 5, lines 1-10 describe Fig. 1 which illustrates a network of clients, a network of printers and a print manager between the networks. While printers are a form of I/O devices, this section does not disclose “determining whether to boost a priority of the application thread according to criteria based on a status of I/O operations performed for the application thread” as claimed.

Col. 6, lines 10-30 describes Fig. 3a which illustrates a boost request data structure, including a Mutex ID field. This section describes the mechanics of boasting the priority of threads but does not “determining whether to boost a priority of the application thread according to criteria based on a status of I/O operations performed for the application thread” as claimed.

In the Response to Arguments section, the Office action again sites to Col. 2, lines 55-65, Fig. 3a and col. 6, lines 10-30 for disclosing “determining whether to boost a priority of

the application thread according to criteria based on a status of I/O operations performed for the application thread.” The Applicant respectfully disagrees.

A close analysis of Fig. 3a and lines 10-30 (describing Fig. 3a). The section describes the mechanics of how to create and manage boost requests. There is no discussion of why threads should be boosted (or not boosted). The missing element is the determination element. There is no discussion of a determination of whether to boost a thread. The section describes how, but not why. There is no discussion of a decision being made as called to in the pending claims. As a result, the claims are a step beyond the disclosure of *Kush*.

The claimed solution is quite elegant and an advancement over the prior art. While *Kush* has to maintain multiple lists and proceed through significant thread priority inheritance reviews, the claimed system focuses on the status of I/O operations performed for the application thread. Examples of when an application thread would be boosted include when no additional I/O operations are in the immediate future, when a given number of I/O operations have already been completed or if a given amount of time has passed since the application thread has been boosted.

As a result of the claimed system, switching between I/O threads and applications threads is accomplished in a way to improve performance. The simplicity of the method reduces overhead in tracking the priority of various threads by focusing on I/O operations which limits the number of threads that need to be analyzed and tracked. This results in improved performance and is a patentable advancement over the prior art.

Claims 2, 11 and 21

Claims 2, 11 and 21 add to their parent claims that “if the step of determining determines not to boost the priority of the application thread, performing a further I/O operation for the application thread, and determining again whether to boost the priority of the application thread.” In other words, if the determination is made that application thread is not to be boosted, repeating the method. The Office action points to col. 10, lines 36-67 for disclosing this element.

Col. 10, lines 36-67 describes Fig. 9 and Fig. 9 is a flowchart. All avenues of the flowchart terminate at a block “End logic.” The method does not repeat as called for in claim 2. Accordingly, claim 2 is not anticipated by *Kush*.

Claims 3, 12 and 22

Claims 3, 12 and 22 add to their parent claims that “wherein the application thread posts a data buffer in connection with the I/O request, and the step of performing the I/O operation includes copying data into the I/O buffer.” The Office action points to column 5, lines 50-55 as disclosing this element. Col. 5, lines 50-55 discloses maintaining a record for each process and threads in the process. The described process record is not the same as an I/O buffer. An I/O buffer holds data while waiting to be delivered to the output device. The described record tracks processes but does not hold I/O data as claimed. Accordingly, this element is not present in *Kush*.

Claims 4, 13 and 23

Claims 4, 13 and 23 add to their parent claims “computer-executable instructions for performing the step of boosting the priority of the application thread when the step of determining determines that the priority of the application is to be boosted.” The Office action points to col. 11, lines 5-15 for disclosing this element. Similar to claim 1, the quoted section does not describe a “determination” but discloses executing a boost request. Accordingly, this element is not present in *Kush*.

Claims 5 and 14

Claims 5 and 14 add to their parent claims that wherein the step of boosting boosts the priority of the application thread by a pre-selected level. The Office action points to Fig. 2 and col. 5, lines 60-67 for disclosing this element. These claims are dependant on their parent claims and elements are not present in the parent claims, these elements are missing in claims 5 and 14 and the claims are not anticipated by *Kush*.

Claims 6, 15 and 23

Claims 6, 15 and 23 add to their parent claims that wherein the pre-selected level is fixed. The Office action points to Fig. 2 and col. 5, lines 60-67 for disclosing this element. These claims are dependant on their parent claims and elements are not present in the parent claims, these elements are missing in claims 5 and 14 and the claims are not anticipated by *Kush*.

Claim 9, 18 and 26

Claim 9, 18 and 26 add to their parent claims that wherein the criteria for determining whether to boost the priority of the application thread include whether a period of time since a last time the priority of the application thread was boosted has reached a threshold length. The Office action points to Fig. 5, and to Col. 8, lines 5-25 for disclosing this element.

Fig. 5 and description thereof in Col. 8, line 5-25 make no mention of time. As there is no mention of time, it is not surprising that there is no mention of a threshold length of time. The concept of time having an affect of a decision to boost a thread is entirely missing from *Kush*. As such, claims 9, 18 and 26 are not anticipated by *Kush*.

Claim Rejections Under 35 USC §103

Claims 7-8, 16-17 and 24-25 are rejected under 35 U.S.C. 103(a) as being unpatentable over *Kush* in view of *Accapadi et al.* US Publication 2005/0022186 (“Accapadi”).

Claim 7, 16 and 24

Claim 7, 16 and 24 call for “wherein the criteria for determining whether to boost the priority of the application thread include whether there are more I/O operations to be performed for the application thread.” *Accapadi* is cited for providing this element in Fig. 1 and paragraph 25. *Accapadi* adds methods to boost certain threads when the threads are about to be blocked but it does not make a determination based on a status of I/O operations performed for the application thread. Fig. 1 illustrates waiting for a critical section to complete. Paragraph 25 is a patent application boilerplate limitation disclaimer. Neither of these citations describe the claimed criteria of determining if there are more I/O operations to be performed. The concept of counting I/O operations is not in the cited sections of *Kush*. As such, these claims are not anticipated and are allowable.

Claim 8, 17 and 25

Claim 8, 17 and 25 call for “wherein the criteria for determining whether to boost the priority of the application thread include whether a number of I/O operations performed in a current thread context for the application thread has reached a threshold number.” *Accapadi*

is cited for providing this element in Fig. 1 and paragraph 25. Accapadi adds methods to boost certain threads when the threads are about to be blocked but it does not make a determination based on a status of I/O operations performed for the application thread. Fig. 1 illustrates waiting for a critical section to complete. Paragraph 25 is a patent application boilerplate limitation disclaimer. Neither of these citations describe the claimed criteria of determining if there are more than a threshold number of I/O operations to be performed. The concept of counting I/O operations is not in the cited sections of *Kush*. As such, these claims are not anticipated and are allowable.

CONCLUSION

In view of the above amendment and arguments, the applicant submits the pending application is in condition for allowance and an early action so indicating is respectfully requested.

The Commissioner is authorized to charge any fee deficiency required by this paper, or credit any overpayment, to Deposit Account No. 13-2855, under Order No. 30835/302629, from which the undersigned is authorized to draw.

Dated: May 21, 2007

Respectfully submitted,

By____/W. J. Kramer/_____
William J. Kramer
Registration No.: 46,229
MARSHALL, GERSTEIN & BORUN LLP
233 S. Wacker Drive, Suite 6300
Sears Tower
Chicago, Illinois 60606-6357
(312) 474-6300
Attorney for Applicant